Chapter 4 Neural Network Based Transformer Fault Diagnosis

Knowledge based power transformer incipient fault diagnosis became popular because of its simplicity, but the application of these standards requires experiences. Most of the time this involves an extensive examination of the gas-in-oil concentrations, and compare the results of several different methods. Expertise is critical in the final phase of drawing a conclusion, because output conflicts often exist between methods. Therefore, knowledge based transformer fault diagnosis is often referred to an art instead of a science, and used to be the "patent" of highly skillful experts.

To approximate those highly skillful experts, this chapter will study some neural network based techniques, compare their performance and discuss their application issues.

4.1 The mechanism of neural network based diagnosis

The basic idea of neural network based fault diagnosis is nonlinear mapping. It is assumed that the relationships between the input vector $\boldsymbol{X}$ and the output vector $\boldsymbol{Y}$ are predefined by the physical nature of the problem, and these relationships can be represented by a limited number of input-output pairs (data samples). These assumptions can raise concerns in some situations, which will be addressed in Section 4.8.

The application of neural network in fault diagnosis has two phases. Phase 1 is the training process, during which the data samples are provided to the network, the memorial coefficients of the network are iteratively adjusted to "memorize" the input-output relationships. Phase 2 is a testing process, during which the input vector $\boldsymbol{x}$ is fed into the network, and the memorized coefficients of the network are recalled to "discover" the possible output.

Phase 1 is usually a bulky processing task and may take many iteration steps to reach the required accuracy (defined in the following sections). This is the phase where application studies often concentrate on, where the network algorithm, topology and input vector can be intentionally changed to optimize the network performance. Once this phase is done, the network specifications are fixed and not to be changed in the testing phase.

Phase 2 is quite straightforward, involves only calculations after the network takes the input vector. The computation time is fairly short with respect to phase 1. Therefore may be possible for real time applications such as on-line transformer fault diagnosis.

4.2 The multi-layer perceptron (MLP) neural network

The MLP is perhaps the most popular neural network used in pattern recognition applications [Hayk99]. As an example, Figure 4-1 shows a two hidden layer MLP, where circles represent neurons, rectangles represent input units, and arrows represent the forward propagated function signals. This is a fully connected network. It has M inputs and N outputs. The memories are weights between the layers and not shown in the figure, but can be represented as w_{ij} in the neuron input-output relationship of Equation 4-1.

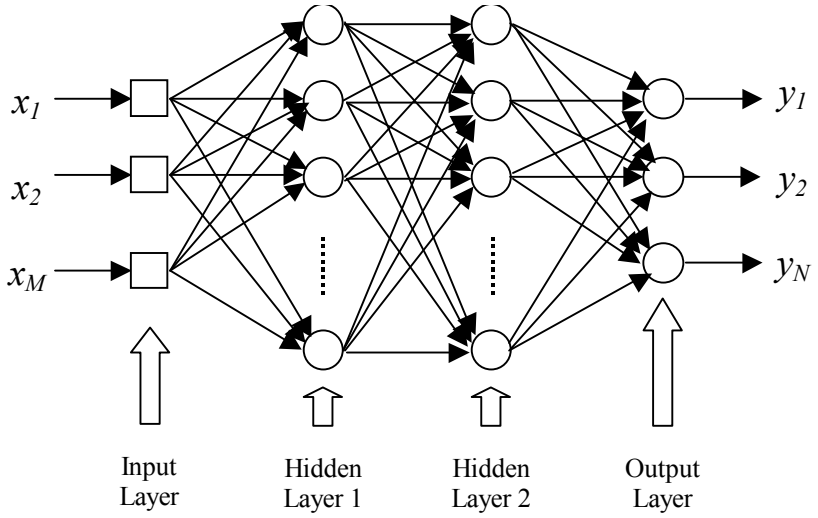

Figure 4-1 Topology of a two hidden layer MLP

$$y_j^{(l)} = \Phi\left(v_j^{(l)}\right) = \Phi\left(\sum_{i=0}^{p} w_{ij}^{(l)} x_{ij}^{(l)}\right) \tag{4-1}$$

where l denotes the layer number ($l > 0$, output layer is the 3rd layer), $y_j^{(l)}$ denotes the output of the jth neuron in the lth layer, $v_j^{(l)}$ denotes the weighted sum of the neuron's inputs, $x_{ij}^{(l)}$ denotes the ith input of the neuron (p inputs from the previous layer and a fixed bias input), $w_{ij}^{(l)}$ denotes the contribution weights of the ith input to the neuron, and $\Phi(\bullet)$ is the activation function of the neuron.

The activation function $\Phi(\bullet)$ is a smooth (i.e. differentiable everywhere) nonlinear function and can have several forms, such as the logistic function of Equation (4-2) and the hyperbolic tangent function of Equation (4-3).

$$\Phi(v) = \frac{1}{1+\exp(-av)} \quad a > 0 \text{ and } -\infty < v < \infty \tag{4-2}$$

$$\Phi(v) = a\tanh(bv) \qquad (a,b) > 0 \tag{4-3}$$

The training of a MLP often uses a back-propagation algorithm, which consists of two passes – the forward pass the backward pass. In the forward pass the weights of the network are fixed and Equation (4-1) is repeatedly used to obtain the outputs from the inputs through all the layers. During the backward pass all the weights are adjusted according to the error-correction equations listed below.

$$e_j(n) = d_j(n) - y_j(n) \tag{4-4}$$

$$\delta_j^{(l)} = \begin{cases} e_j^{(L)}\Phi'\left(v_j^{(L)}(n)\right) & \textit{for neuron } j \textit{ in output layer } L \\ \Phi'\left(v_j^{(l)}(n)\right)\sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) & \textit{for neuron } j \textit{ in hidden layer } l \end{cases} \tag{4-5}$$

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha w_{ji}^{(l)}(n-1) + \eta\delta_j^{(l)}(n) y_i^{(l-1)}(n) \tag{4-6}$$

where n is the training iteration number, e is the error signal, d is the required signal, $\Phi'(\bullet)$ denotes a differentiation, η is the learning-rate parameter and α is the momentum constant. The selection of η and α will be discussed later in this chapter.

During the training process, data samples should be presented to the network randomly. Presenting all the data samples to the network once is named as an epoch. Many epochs are usually necessary to train a network. The training ends when the squared individual errors and/or averaged system error are less than the preset values. These errors are defined as:

$$e_{squared} = \frac{1}{2}\left[e_j(n)\right]^2 \tag{4-7}$$

$$e_{averaged} = \frac{1}{2N}\sum_{j=1}^{N}\left[e_j(n)\right]^2 \tag{4-8}$$

The selection of a MLP in this study has profound basis. First of all, the transformer fault diagnosis problem is likely a highly complex nonlinear mapping problem because both the inputs and outputs are multiple variables and there is no linear relationship has ever been found. Secondly, even a three layer MLP (with one hidden layer) has been proved to have the capability of approximating any function regardless of its complexity, MLPs with more than one hidden layer should be more powerful. Thirdly, MLPs with a supervised error back-propagation (BP) training algorithm have been applied successfully to solve some difficult and diverse problems. It was hope that a MLP could meet all our needs.

However, the application of MLP involves many issues, as will be addressed in Section 4.5, 4.6 and 4.7. MLPs with multiple outputs or multiple hidden layers were proved to be unsuitable for the job. On the other hand, a single-hidden-layer single-output MLP was identified as the best building block of a modular neural network, whose topology will be given in Section 4.4. Before that is another type of neural network studied but discarded after performance comparison.

4.3 The learning vector quantization (LVQ) neural network

Strictly speaking the learning vector quantization (LVQ) neural network is not a neural network but a supervised learning process of a self-organizing feature map [Hayk99, pp467]. It is so named only because it is a separate sub-category in neural network classifications.

Figure 4-2 is the topology of a LVQ network. It has three layers: the input layer, the competitive layer, and the output layer. The neurons in the competitive layer are divided into *n* groups. Each group has the same number of neurons and corresponds to an output layer neuron. Classification information is stored in the weight matrix $\boldsymbol{W}_{x\text{-}c}$ fully connected between input layer neurons and competitive layer neurons.

During the learning process, the Euclidean distances between an input vector $\boldsymbol{x}_k$ and the weights of each neuron in the competitive layer are calculated, the neuron with the minimum distance is labeled as a winner and the corresponding output neuron is activated. Assume neuron *j* is the winner, $\boldsymbol{w}_j$ is the corresponding weight vector. If the output is correct, $\boldsymbol{w}_j$ moves towards the input vector,

$$\boldsymbol{w}_j(n+1) = \boldsymbol{w}_j(n) + \alpha_n[\boldsymbol{x}_k - \boldsymbol{w}_j(n)] \tag{4-9}$$

Otherwise they are moved away from the input vector,

$$\boldsymbol{w}_j(n+1) = \boldsymbol{w}_j(n) - \alpha_n[\boldsymbol{x}_k - \boldsymbol{w}_j(n)] \tag{4-10}$$

Where $\alpha_n \in (0,1)$ should decrease with the training step n.

Classification process is similar to the first part of the learning process. No weight adjustment is made and only the output pattern is determined from distance calculation and comparison.

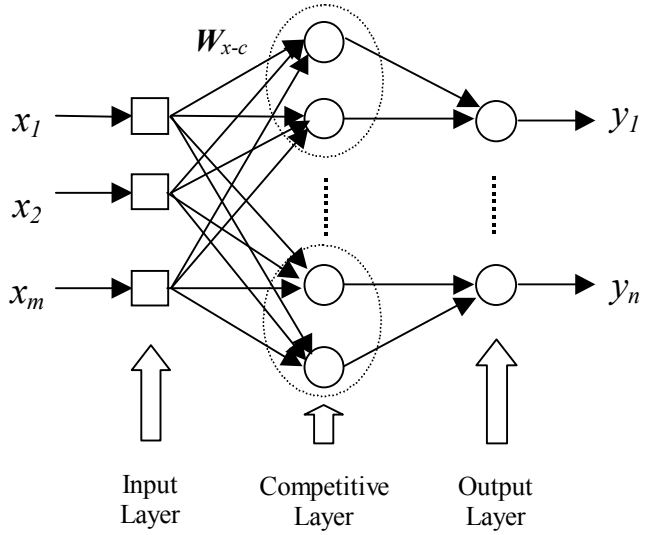

Figure 4-2 Topology of the LVQ networks

During the winner determination process, other comparison parameter may be used besides the direct Euclidean distance. For example, the likelihood function may be used.

$$p_i = \sum_{q=1}^{S} \exp\left(-\frac{d_q^2}{2C^2}\right) \qquad i = 1, 2, \ldots, N \tag{4-11}$$

Where S is the number of neurons in the competitive layer for each output neuron, d_q is the Euclidean distance between input vector to the q-th neuron of the S neurons, C is a constant, p_i is the likelihood of the input vector to the i-th output neuron. The output neuron is selected corresponding to the greatest value of p_i.

The principle of LVQ network is very similar to that of a Nearest-Neighbor Rule (NNR) classifier and a Multivariate Gaussian (MVG) classifier. They all use some typical samples to

represent a class of samples. These representing samples are the global or local center of the class cluster. In a NNR classifier or a LVQ network there could be one or more such samples for one class, while in a MVG classifier there is only one representing sample for each class.

The difference between a NNR classifier and a LVQ network is that the former has real representing samples obtained from Editing and Condensing algorithms (Appendix 3), while the later has pseudo representing samples that are combined versions of many training samples through LVQ algorithms. Intuitively, they are equivalent to each other in terms of classification performance.

Although the MVG classifier only has one representing sample for each class, it uses statistical methods to improve the representation capability of the sample, therefore its overall classification performance is not necessarily lower than a NNR classifier or a LVQ network. The MVG classifier is described in Appendix 4.

4.4 The modular neural network

Modular networks are introduced to deal with very complex function approximation and/or pattern recognition problems [Kayk99]. In a modular network, the computation tasks are decomposed into two or more modules (subsystems) that operate on distinct inputs without communicate with each other. The outputs of the modules are mediated by an integrating unit, which is not permitted to feed information back to the modules.

For a multiple-input single-output problem, the modular network could be of a committee machine show in Figure 4-3, where the combiner takes care of the modular output integration task using ensemble-averaging technique.

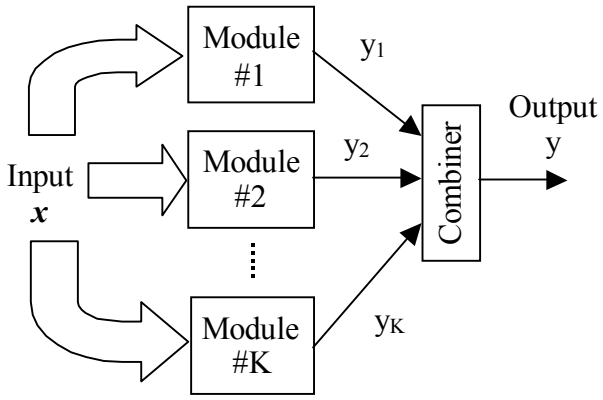

Figure 4-3 A committee machine type modular network

For the multiple-input multiple-output pattern recognition problem in this study, the simple network of Figure 4-3 is not suitable in that the concern here is not to combine the outputs, but to assign proper input vectors to different modules. The topology of modules could be different, too. This involves an extensive study of input vector optimization for each module and the topology optimization of each module. Section 4.5 and 4.6 will summarize the results. Figure 4-4 gives the modular network structure shown to be suitable from the study.

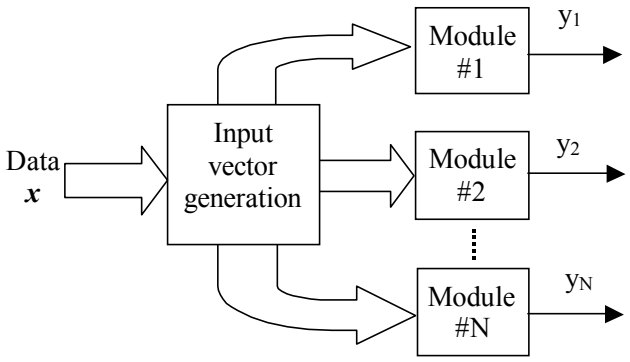

Figure 4-4 The modular network used for transformer fault diagnosis

4.5 Optimization of input vectors

This is to find a better feature space for the fault diagnosis tasks. It is closely related to neural network optimization in Section 4.6.

The work can be traced back to Zhang's study at Virginia Tech [Zhang96]. Based on 40 data samples, multiple output MLPs and a ten-fold-cross-validation technique, he concluded that an input vector with H_2, CH_4, C_2H_2, C_2H_4 and C_2H_6 concentrations is the best choice for major fault (overheating, corona and arcing) diagnosis. For cellulose degradation diagnosis, the best choice of input vector consisting of H_2, CH_4, C_2H_2, C_2H_4, C_2H_6, CO and CO_2 concentrations.

This study is a continuation of Zhang's work. The purpose was to validate the above conclusions. It was based on 77 data samples gathered from Doble Engineering Company and other sources (publications). Each sample contains gas-in-oil concentrations of H_2, CH_4, C_2H_2, C_2H_4, C_2H_6, CO and CO_2, and the transformer condition. The 77 samples include 11 normal cases, 41 overheating cases, 4 corona cases, 16 arcing cases and 5 overheating/ arcing cases. Since the corona cases were too little, they merged into arcing cases and were renamed as discharge cases.

The 77 data samples were divided into two data sets, the training data set TRN_IPO and the testing data set TST_IPO. TRN_IPO contains 67 data samples. TST_IPO has 10 data samples. They are listed in Appendix 5.

Two types of input vectors were compared in the study. One had CO and CO_2 concentrations and the other had not, but both had concentrations of H_2, CH_4, C_2H_2, C_2H_4 and C_2H_6. The concentrations were divided by 1000 before entering input vectors. Other data scaling methods such as normalization and ratio calculation of the concentrations were also studied but failed to improve the training speed and testing accuracy.

To save training and testing time, which is necessary when the number of data samples is large and the complexity of the network is high, a new method was used to evaluate the performance of the network instead of the ten-folder cross-validation technique. In this method, the testing accuracy A_{tst} is an average of several training-testing runs:

$$A_{tst} = \frac{1}{N_{tst-run}} \sum_{i=1}^{N_{tst-run}} A_{tst-run}(i) = \frac{1}{N_{tst-run}} \sum_{i=1}^{N_{tst-run}} \frac{1}{N_{tst}} \sum_{j=1}^{N_{tst}} A_{tst}(i,j) \qquad (4\text{-}12)$$

Where $N_{tst-run}$ is the number of training-testing runs, N_{tst} is the number of data samples in the testing data set, $A_{tst-run}(i)$ is the average testing accuracy of the *i*th training-testing run, $A_{tst}(i,j)$ is the individual testing accuracy of data sample *j* at the *i*th training-testing run. If the individual test gives correct fault classification then $A_{tst}(i,j)$=1, otherwise $A_{tst}(i,j)$=0.

Many MLP topologies were studied and the testing accuracies (A_{tst}, in percentage) are shown in Table 4-1 and 4-2, where M is the number of neural network input features and "MLP topology" indicates the number of layers and nodes (neurons) in each layer. For example, a MLP topology of "M-2M-3" means that the MLP has three layers, and the node number for its input layer, hidden layer and output layer is M, 2M and 3, respectively.

When a MLP has 3 nodes in the output layer, the 3 outputs represent "normal", "overheating" and "discharge" condition, respectively. When a MLP has only one output (single output), the output represents either "normal", or "overheating", or "discharge" conditions.

The following observations are from Table 4-1 and 4-2.

Table 4-1 Testing accuracy (%) of 3-output MLPs with/without CO, CO_2 concentrations in the input vectors

MLP Topology	Normal		Overheating		Discharge	
	M = 5	M = 7	M = 5	M = 7	M = 5	M = 7
M-2M-3	95	83.75	80	85	80	85
M-3M-3	95	87.5	90	91.25	87.5	86.25
M-4M-3	93.75	86.25	87.5	85	85	80
M-5M-3	95	87.5	85	88.75	85	83.75
M-3M-9-3	93.75	91.25	85	85	87.5	82.5
M-4M-12-3	83.75	85	83.75	81.25	87.5	78.75
M-5M-15-3	95	85	85	78.75	88.75	72.5
M-40-20-3	92.5	85	82.5	88.75	87.5	80
Average	93	86.4	84.8	85.5	86.1	81.1

Table 4-2 Testing accuracy (%) of single-output MLPs with/without CO, CO_2 concentrations in the input vectors

MLP Topology	Normal		Overheating		Discharge	
	M = 5	M = 7	M = 5	M = 7	M = 5	M = 7
M-2M-1	96.25	88.75	88.75	86.25	87.5	86.25
M-3M-1	96.25	88.75	88.75	83.75	88.75	83.75
M-4M-1	96.25	91.25	90	85	88.75	80
M-5M-1	96.25	88.75	87.5	85	88.75	82.5
Average	96.25	89.4	88.75	85	88.44	83.13

With 3-output MLP topologies, the benefit of CO and CO_2 absence in the input vectors is not obvious for "overheating" diagnosis but quite clear for "normal" and "discharge" diagnosis. These partly agree with the conclusions of [Zhang96].

With single-output MLP topologies, however, higher testing accuracies of "normal", "overheating" and "discharge" diagnosis are clearly related to the absence of CO and CO_2. This not only agrees with [Zhang96] but also has special meaning and will be further addressed in the next section.

4.6 Neural network optimization

This refers to the selection of a better neural network type and topology, which involves four phases. The first phase is to identify the optimal MLP topology. The second phase is to identify the optimal LVQ network topology. The third phase is to study the MVG classifiers. The fourth phase is selecting the best from these three types of pattern recognition tools.

4.6.1 Optimal MLP topology identification

Data sets TRN_IPO and TST_IPO (Appendix 5) were used in the first phase, and Equation 4-12 was used to evaluate the MLP performance. Table 4-3 and 4-4 show the results.

Table 4-3 lists three types of MLP topologies: single-output one-hidden-layer MLPs, multiple-output one-hidden-layer MLPs and multiple-output two-hidden layer MLPs, where M = 5 is the number of input features (dimension of input vectors). The overall average classification accuracy for these three types of MLP topologies is 91.1%, 88.2% and 87.7%, respectively. It is obvious that the first type of MLP topology, i.e. the single-output one-hidden-layer MLP, is the best choice.

Table 4-4 lists two types of MLP topologies for cellulose degradation diagnosis. They are all one-hidden-layer MLPs but have different output numbers. The single-output MLPs are obviously better than the multiple-output ones.

According to Table 4-3 and 4-4, single-output one-hidden-layer is selected as the optimal MLP topology. This actually implies that the modular neural network structure of Figure 4-4 is chosen. This partly solved the optimization problem because it is still hard to find the best topology from the single-output one-hidden-layer MLPs that were studied. It seems like the number of hidden

layer neurons does not have too much impact on the MLP performance. To solve the optimization problem completely, engineering judgment must be used and will be discussed in Section 4.7.

Table 4-3 MLP based condition classification accuracy (%) of data set TST_IPO

MLP Topology	Normal	Overheating	Discharge	Average
M-2M-1	96.25	88.75	87.5	90.83
M-3M-1	96.25	88.75	88.75	91.25
M-4M-1	96.25	90	88.75	91.67
M-5M-1	96.25	87.5	88.75	90.83
M-2M-3	95	80	80	85
M-3M-3	95	90	87.5	90.83
M-4M-3	93.75	87.5	85	88.75
M-5M-3	95	85	85	88.33
M-3M-9-3	93.75	85	87.5	88.75
M-4M-12-3	83.75	83.75	87.5	85
M-5M-15-3	95	85	88.75	89.58
M-40-20-3	92.5	82.5	87.5	87.5

Table 4-4 Testing accuracy (%) of MLP based cellulose degradation diagnosis for data set TST_IPO

MLP Topology	7-14-N	7-21-N	7-28-N	7-35-N
N = 4	56.25	63.75	67.5	65
N = 1	73.75	72.5	71.25	65

4.6.2 LVQ network study

The study of LVQ networks and MVG classifiers is based on two new data sets, TRN_DBL1 and TST_DBL1 (see Appendix 6 for details), which are all provided by Doble Engineering Company and validated by highly skilled DGA experts. The fault conditions of these data samples are classified into five categories (also see Appendix 6 for details):

- Overheating regardless of oil or cellulose (OH)
- Overheating of Oil (OHO)
- Low Energy Discharge (LED)
- High Energy Discharge or Arcing (HEDA)
- Cellulose Degradation (CD)

Each of the studied LVQ network has two outputs. One stands for an individual fault condition and the other stands for the complementary of this fault condition. Assume N_I is the dimension of the input vector, N_C is the neuron number of the competitive layer, Table 4-5 and 4-6 shows the testing results of five training-testing runs. N_I = 7 means that the seven major gas-in-oil concentrations form the input vector. N_I = 5 means that CO and CO_2 are excluded from the input vector.

Table 4-5 Average testing accuracy (%) of studied LVQ networks

SET	TRN_DBL1								TST_DBL1							
N_C	4		6		8		10		4		6		8		10	
N_I	7	5	7	5	7	5	7	5	7	5	7	5	7	5	7	5
OH	75.1	74	80	78.3	80.5	70.3	**82**	60.3	77.2	63.6	80.4	67.2	78.4	66.4	**77.6**	65.6
OHO	75.2	85.5	79.2	84.7	76.7	90.9	79.2	**91.1**	73.2	84.4	78.4	84.4	76.4	88	78.4	**87.6**
LED	-	-	89.2	**90.9**	89.3	90.3	89.7	90.1	-	-	85.2	**86.4**	86	86.8	84.8	87.2
HEDA	76.4	80.4	81.3	80.1	76.1	**82.3**	78.9	82.1	74.4	72.8	78	73.6	71.6	**76.8**	72.4	75.6
CD	28.5	-	28.8	-	**28.8**	-	28.4	-	62	-	64.8	-	**66**	-	64.8	-

Table 4-6 Standard deviation (%) of testing accuracies for the studied LVQ networks

SET	TRN_DBL1								TST_DBL1							
N_C	4		6		8		10		4		6		8		10	
N_I	7	5	7	5	7	5	7	5	7	5	7	5	7	5	7	5
OH	5.9	1.1	2.8	9.1	2.1	12.5	2.4	15.6	2.3	4.6	3.3	5.6	3.6	8.6	4.1	7.5
OHO	5.4	1.7	3.8	2.0	4.7	0.7	4.4	0.8	5.4	0.9	3.3	0.9	5.5	1.4	7.7	1.7
LED	-	-	2.9	0.4	0.7	0.4	1.1	0.6	-	-	1.1	0.9	1.4	1.1	1.8	1.1
HEDA	5.5	1.0	1.7	1.1	4.7	1.4	6.5	1.1	3.6	1.8	1.4	2.2	3.8	1.8	7.3	2.2
CD	1.2	-	0.9	-	1.9	-	1	-	0	-	2.3	-	0	-	1.1	-

From Table 4-5 we have the following observations. First, LVQ networks are not suitable for cellulose degradation (CD) diagnosis, because the testing accuracy is very low. Second, the dimension of input vectors may greatly affect the performance of a LVQ network. For overheating (OH) diagnosis, 7 inputs are better than 5 inputs and the testing accuracy can be improved significantly (about 20% when N_C = 10). For overheating of oil (OHO) diagnosis, the testing accuracy may increase over 10% if 5 inputs are used instead of 7 inputs. For low-energy discharge (LED) and high-energy discharge (HEDA) diagnosis, 5 inputs are better than 7 inputs to some extent but the testing accuracy improvement is not significant, being about 1% and 5%, respectively. Third, neuron number of competitive layer affects the LVQ network performance but not too much by only a few percent. The best LVQ networks are marked in Table 4-5 according to their testing accuracy (in bold).

A comparison between Table 4-5 and 4-6 reveals the correlation between LVQ testing accuracies and their standard deviation. That is, when the standard deviation is low, the testing accuracy is usually high. This is in favor of the selected LVQ topologies because a smaller standard deviation means that the testing accuracies are more concentrated around their numerical center and the LVQ network's training is less sensitive to its initialization, therefore is more robust in its performance. An exception is the CD diagnosis, where the testing accuracy is low but the standard deviation is also very low. An explanation could be that the LVQ network is really not suitable for CD diagnosis since there is no chance for it to reach a high testing accuracy.

4.6.3 MVG classifier study

The priori probability is the only parameter that needs to be determined for a MVG classifier and there are two ways of doing so. The first is to estimate the probability from the training data set. The other is to set it intuitively (by default, a priori probability of 0.5 can be assigned to each individual fault type). Both methods were tried in the study and the results are given in Table 4-7, where p_{fault} is the priori probability of the individual fault type, N_I is the dimension of the input vector.

Table 4-7 Testing accuracy (%) of studied MVG classifiers

Data Set	p_{fault}	N_I	OH	OHO	LED	HEDA	CD
TRN_DBL1	0.5	7	93.3	90.0	94.0	**98.7**	**95.3**
		5	38.7	82.7	8.0	28.0	-
	Estimated from TRN_DBL1	7	91.3	**95.3**	**95.3**	97.3	93.3
		5	**94.0**	26.0	7.3	23.3	-
TST_DBL1	0.5	7	58.0	56.0	74.0	**84.0**	**44.0**
		5	48.0	80.0	6.0	26.0	-
	Estimated from TRN_DBL1	7	52.0	**90.0**	**82.0**	86.0	36.0
		5	**98.0**	26.0	2.0	18.0	-
TRN_DBL1 + TST_DBL1	0.5	7	84.5	81.5	89.0	**95.0**	**82.5**
		5	41.0	82.0	7.5	27.5	-
	Estimated from TRN_DBL1	7	81.5	**94.0**	**92.0**	94.5	79.0
		5	**95.0**	26.0	6.0	61.0	-

In Table 4-7, the highest testing accuracies for each individual fault diagnosis are marked with bold font. It is clear that except for CD diagnosis, using the priori probability estimated from the training data set TRN_DBL1 yields better classifier performance. On the other hand, a 5 inputs MVG classifier is better for OH diagnosis and a 7 inputs MVG classifier is better for OHO, LED and HEDA diagnosis, which is just the opposite of LVQ network study observations. The reason

is probably related to the nature of MVG classifiers and also the selected data set. We will not go deep into this but will use the results for comparison in the next section.

4.6.4 Selection of the best fault diagnosis tool

In order to select the best from the three types of fault diagnosis tools studied in this chapter, Table 4-8 is generated, where the testing accuracies for LVQ and MVG are summarized from Table 4-5 and 4-7. The MLP networks have the topology determined in Section 4.6.1, i.e. they are all single-output one-hidden-layer MLP.

Table 4-8 Testing accuracy (%) of three tools on data set TRN_DBL1 and TST_DBL1

Fault Type		OH		OHO		LED		HEDA		CD	
Data Set	Tools	N_I	%	N_I	%	N_I	%	N_I	%	N_I	%
TRN_DBL1	MLP	7	100	5	100	5	100	5	98	7	98.7
	MVG	5	94	7	95.3	7	95.3	7	98.7	7	95.3
	LVQ	7	82	5	91.1	5	90.9	5	82.3	7	28.8
TST_DBL1	MLP	7	94	5	94	5	94	5	88	7	92
	MVG	5	98	7	90	7	82	7	84	7	44
	LVQ	7	77.6	5	87.6	5	86.4	5	76.8	7	66
TRN_DBL1 + TST_DBL1	MLP	7	98.5	5	98.5	5	98.5	5	95.5	7	97
	MVG	5	95	7	94	7	92	7	95	7	82.5
	LVQ	7	80.9	5	90.2	5	89.8	5	80.9	7	38.1

Table 4-8 basically tells us that MLP is the best among the three types of pattern recognition tools. It did really well in all the individual fault diagnosis of both the training data set TRN_DBL1 and the testing data set TST_DBL1. MVG classifiers almost matched up with MLP networks in general, even did better in some cases (for HEDA diagnosis of TRN_DBL1 and OH diagnosis of TST_DBL1), but did really bad for CD diagnosis of TST_DBL1. LVQ networks work fine for OHO and LED diagnosis, fair for OH and HEDA diagnosis, but very bad for CD diagnosis.

It should be noted that from Section 4.6.2 to 4.6.4 the fault type OH is slightly different from that of Section 4.5 and 4.6.1 in that it also counts overheating of cellulose material. This does affect the choice of input vector dimension for MLP based OH diagnosis. Seven instead of five is chosen as the optimal dimension.

4.7 MLP applications issues

Following are some important issues related to the application of MLP networks in power transformer fault diagnosis. They are the summary of experiences gained from the study.

4.7.1 Selection of the training data set

The training data set affects the speed of training and the performance of a neural network. Under the assumption that the fault diagnosis problem is a boundary searching process, inconsistence in the training data set, i.e. confusion of data samples around the boundary, could make the neural network very hard to train. If the training convergence limit is set too flexible, the testing accuracy may well below the acceptable level. If the training convergence limit is set too strict, the training process may last a very long time and over-training likely occurs, where the testing accuracy of the training data set is high but it is low for the testing data set.

Elaborate selection of the training data set could yield good results. In the MLP topology optimization study of Section 4.6.1, the training data set TRN_IPO consists of data samples from different sources and they are likely not consistent with each other. In the pattern recognition tool comparison study of Section 4.6.4, the training data set TRN_DBL1 consists of data samples from only one source and they likely have good consistence. If we compare the MLP testing accuracy of Table 4-3 and 4-8, we can see that TRN_DBL1 does act better than TRN_IPO.

4.7.2 MLP training

In the training of a MLP, two parameters must be properly selected to ensure fast training and convergence. These are the learning rate η and the momentum constant α in Equation (4-6). Through extensive study Zhang concluded that $\eta = 0.3$ and $\alpha = 0.7$ is good selection for the fault diagnosis problem of power transformers [Zhang96]. In this study the same parameters were used and the results confirmed Zhang's conclusion. No further study was conducted to further investigate the issue.

As mentioned in the last section, over-training could be a problem for multiple-source training data set. There are two types of solution for the problem. One is early termination of the training process like setting a flexible training limit, but the side effect of this solution could be a low testing accuracy for both the training data set and the testing data set. The other is elaborately select the data samples of the training data set, make sure they are consistent in stead of conflict with each other, which could result in fast training and high testing accuracy. In this study, TRN_DBL1 was the final choice to train the MLPs used in the fault diagnosis.

4.7.3 MLP topology optimization

An optimal MLP topology was the goal of many researchers. In the field of neural network based power transformer fault diagnosis, Ding and Zhang both found an MLP topology [Ding95, Zhang97] based on their training data set. But how they concluded the "optimal" needs to be analyzed.

Intuitively if the MLP training convergence fast and the final error is low, the topology would be a good one. Ding and Zhang's optimal topology were basically based on this idea. However, fast training convergence reflects high degree of consistency in the training data set and it does not necessarily mean a high testing accuracy, especially when the training data set is not large enough to be representative. Therefore it is not sufficient to use fast training convergence as the only criteria to judge a MLP topology is optimal or not. In conclusion, the optimal MLP topology for DGA based power transformer fault diagnosis should have both a fast training convergence and a high testing accuracy, with an emphasis on the latter.

In practice, a power transformer fault diagnosis system is likely an off-line system and the MLP networks do not need to be trained in real time, therefore a very fast training convergence is not critically necessary. On the other hand, if an on-line power transformer fault diagnosis system is used, it does require the MLP to be trained on-line when additional data samples become available. Its topology must ensure that it can be trained to a preset residual error level within a reasonable time frame.

Zhang used a ten-fold cross-validation method to evaluate the performance of the studied MLP. When the number of data samples is small, say less than 50, there are some advantages of doing

so. For instance, the data scaling method and the preliminary optimal MLP topology can be studied with a limited number of data samples.

However, the ten-fold cross-validation method implies that all the data samples are perfectly representatives of the problem. If some of them are not so sure about their desired responses, the method may mislead the optimal MLP topology conclusion. The power transformer fault diagnosis problem is unfortunately just such a problem, because even an expert cannot be sure he is 100% correct with his conclusion in many cases. Based on this understanding, extra care must be taken to validate the data before using them.

When the number of data samples is large, ten-fold cross-validation method is not a practical method, and Equation 4-2 is a good alternative. Careful check on the data samples and selection of the training data set are a must before using it.

Even if all the previous steps were taken properly, the MLP topology optimization is still not done. Some topologies may appear to be at the same competitive level. At this moment, engineering judgment is necessary to decide which one is "optimal". The basic consideration is that large number of hidden layer neurons can improve the MLP's diagnostic accuracy to some extent by increasing its complexity, but the improvement may not justify the addition of required resources (memory and processing time).

After examination of Table 4-3 and 4-4 and applying engineering judgment, M-3M-1 type MLP is selected to be the optimal MLP topology for power transformer fault diagnosis. This MLP topology was used in the generation of MLP testing accuracies in Table 4-8. We already saw that they are good.

4.8 Summaries

This chapter studied neural network based power transformer fault diagnosis, and concluded that three-layer single-output MLP is the best choice.

First the principle of neural network based fault diagnosis was introduced. Then several types of neural networks were reviewed, including the multi-layer perceptron (MLP), the learning vector quantization (LVQ) and the modular network.

The studies include input vector optimization and network topology optimization. The input vector optimization was based on 77 data samples from several sources, and it concluded that for

"normal", "overheating" and "discharge" diagnosis, five gas-in-oil concentrations including H_2, CH_4, C_2H_2, C_2H_4 and C_2H_6 are the best choice, and this coincides with previous study on this issue. The network topology optimization was intensively studied based on two groups of data set. The first group has 77 data samples and the second group has 200 data samples. Besides the MLP, multivariate Gaussian (MVG) classifiers and learning vector quantization (LVQ) networks were also studied for performance comparison. It was concluded that MLP based modular network topology is the best choice.

Some MLP application issues were addressed, including the selection of data set and MLP training parameters, over training solution, MLP performance evaluation, and how to select the optimal MLP topology.